WHAT IS CLAIMED IS:

1.   A parallel loop transformation method for race detection during an execution of parallel programs in which the original parallel loop is transformed into a full race covering loop at race detection during the program execution subjected to parallel loop programs, comprising the steps of:

(a) generating the data structure of a condition statement branch determinant string Cstr required for loop transformation taking the parallel loop as an input and extracting the execution path information;

(b) transforming the parallel loop into a full race covering loop using said data structure of a condition statement branch determinant string Cstr required for loop transformation and said execution paths information;

(c) instrumenting the race detection function in order to activate race detection function for the transformed parallel loop which are generated at said step (b); and

(d) executing race detection while running the parallel program according to instrumented detection functions which are determined at said step (c).

2.   The method as claimed in Claim 1, wherein said step (a) further comprises the steps of:

(a-1) sequentially receiving each statement of each parallel loop body in order to generate a single Cstr data structure for each single parallel loop;

(a-2) assigning a bit variable which can store a true or false value to corresponding if-statement if said input statement is an if-statement; and

(a-3) extracting the Cstr data structure and the number of execution path for each parallel loop through an arbitrary path analyzer after assigning said bit variable.

12

3. The method as claimed in Claim 1, wherein said step (b) further comprises the steps of:

(b-1) determining whether the input statement is the first statement or not after a

5 new statement is inputted to the loop body;

(b-2) inserting an execution path control statement, prior to the input statement if the inputted statement is determined to be the first statement, which dynamically assigns an appropriate value for Cstr in order to allow each iteration to have an intended execution path so as to minimize the duplicate monitoring for race detection against the parallel

10 loop;

(b-3) substituting the conditional equation C1 of a conditional statement by a conditional statement for $(Cstr[c\_con\_bit].eq.1) \wedge ((\neg C1) \vee (C1))$ if the present statement is determined to be a conditional statement after the execution path control statement is inserted; and

15 (b-4) repeating the above actions until the inputted statement is determined to be the last statement where, if the present statement is not an if-statement, the input statement is maintained as it is and the above processes are repeated until a parallel loop is transformed into full race covering loop.

20 4. The method as claimed in Claim 3, wherein said present statement is not an if-statement, the input statement is maintained as it is and the above processes are repeated until a parallel loop is transformed into full race covering loop.

5. The method as claimed in Claim 3, wherein said execution path control

25 statement can be listed as;

13

p = (I/stride) mod k;

if (p .eq. 0)

   Cstr = (0)2

Else if (p .eq. 1)

    Cstr = (1)2;

Else if (p .eq. K-1)

Cstr = (K-1)2;

Endif

, where I is loop control variable and k is the number of paths.

6.   The method as claimed in Claim 3,   , wherein said execution path control statement determines the value of Cstr which is to be used for determining the execution path of the loop body from the present iteration using the value of the present loop control variable of each iteration.

7.   The method as claimed in Claim 3, wherein said substituted conditional statement determines the branching of the present conditional statement using the Cstr value corresponding to the present conditional equation while maintaining the semantic of the original conditional equation.

8.   The method as claimed in Claim 1, wherein said step (c) further comprises the steps of:

(c-1) determining whether the statement inputted to instrument an appropriate race detection function for the transformed parallel loop is the beginning and ending statement of the parallel loop;

14

(c-2) instrumenting a label creation statement and a end statement which function on the iteration less than two times of the front and end execution paths until the last statement is inputted if the inputted statement is determined to be either the beginning and ending statement of the parallel loop;

5

(c-4) instrumenting the inspection statement, which inspects whether or not the accessing incident participates in the race, functions on the iteration less than two times of the execution paths until the last statement if the present statement includes an accessing incident of the shared variables.

10

9. The method as claimed in Claim 8, where said step (c) further comprises the steps of:

(c-5) inspecting the shared variables of said present statement to see whether any accessing incident is included if said input statement is neither the beginning nor the

15   ending of a parallel loop; and

(c-6) instrumenting the statement which inspects the inclusion of these accessing incidents to function on the iteration less than two times of the execution paths if the shared variables of the present statement include accessing incidents.

20   10. A recording method of the recording medium which implements all the executable instructions of a digital processing apparatus according to their types comprises and can be written/read from said digital processing apparatus, comprising the steps of:

(a) generating the data structure of a condition statement branch determinant string

25   Cstr required for loop transformation taking the parallel loop as an input and extracting

15

the execution path information;

(b) transforming the parallel loop into a full race covering loop using said data structure of a condition statement branch determinant string Cstr required for loop transformation and said execution paths information;

5   (c) instrumenting the race detection function in order to activate race detection within the iteration instances necessary for the transformed parallel loop which are generated at said step (b); and

(d) executing race detection while running the parallel program according to instrumented detection functions which are determined at said step (c).

10

16